1PW

OIPE IAP1
MAY 19 2006
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Eric Plante	Examiner:	Peter Pappas
Serial No.:	10/622,009	Group Art Unit:	2628
Filed:	July 17, 2003	Docket:	G&C 30566.336-US-01
Title:	PROCESSING SCENE OBJECTS		

CERTIFICATE OF MAILING OR TRANSMISSION UNDER 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on May 16, 2006.

By: [signature]
Name: Jason S. Feldmar

MAIL STOP ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

We are transmitting herewith the attached:

☒ Transmittal sheet, in duplicate, containing a Certificate of Mailing under 37 CFR 1.8.
☒ Communication Regarding Priority Document.
☒ Certified copy of a UNITED KINGDOM application, Serial No. 0216839.1, filed July 19, 2002, the right of priority of which is claimed under 35 U.S.C. 119.
☒ Return postcard.

Please consider this a PETITION FOR EXTENSION OF TIME for a sufficient number of months to enter these papers, if appropriate.

Please charge all fees to Deposit Account No. 50-0494 of Gates & Cooper LLP. A duplicate of this paper is enclosed.

Customer Number 22462
GATES & COOPER LLP
Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, CA 90045
(310) 641-8797

By: [signature]
Name: Jason S. Feldmar
Reg. No.: 39,187
JSF/bjs

(PTO TRANSMITTAL - GENERAL)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Eric Plante	Examiner:	Peter Pappas
Serial No.:	10/622,009	Group Art Unit:	2628
Filed:	July 17, 2003	Docket:	G&C 30566.336-US-01
Title:	PROCESSING SCENE OBJECTS		

CERTIFICATE OF MAILING OR TRANSMISSION UNDER 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on May 16, 2006.

By: Jason S. Feldmar
Name: Jason S. Feldmar

COMMUNICATION REGARDING PRIORITY DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Please place the following Certified Priority Document into the file of the above-identified patent application, as follows:

UNITED KINGDOM, Application No. 0216839.1, Filed July 19, 2002

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: May 16, 2006

By: Jason S. Feldmar
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/bjs

G&C 30566.336-US-01

The Patent Office

For Innovation

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation and Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein together with the Statement of inventorship and of right to grant of a Patent (Form 7/77), which was subsequently filed.

I also certify that the attached copy of the request for grant of a Patent (Form 1/77) bears an amendment, effected by this office, following a request by the applicant and agreed to by the Comptroller-General.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

[illegible]ccordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P[illegible] or PLC.

Re-re[illegible]ation under the Companies Act does not constitute a new legal entity but merely subjec[illegible] company to certain additional company law rules.

Signed

Dated 5 May 2006

19JUL02 E734734-1 003028
P01/7700 0.00-0216839.1

THE PATENT OFFICE
A
19 JUL 2002
RECEIVED BY FAX

Patents Form 1/77

Patents Act 1977

The Patent Office

1/77

Request for grant of a patent

Grant

The Patent Office
Concept House
Cardiff Road
Newport
Gwent NP10 8QQ

1. Your reference — 2034-P585-GB

2. Pa... — 0216839.1 — NEW 19 JUL 2002

... name, address and postcode of the or of each applicant *(underline all surnames)*

AUTODESK CANADA INC
10 Duke Street
Montreal
Quebec
Canada H3C 2L7

Patents ADP number *(if you know it)* — 637806901

If the applicant is a corporate body, give the country/state of its incorporation — Quebec, Canada

4. Title of the invention — PROCESSING SCENE OBJECTS

5. Name of your agent — ~~ATKINSON BURRINGTON~~

"Address for service" in the United Kingdom to which all correspondence should be sent

~~25-29 President Buildings~~
~~President Way~~
~~Sheffield S4 7UR~~
GB

Marks & Clerk
57-60 Lincoln's In Fields
London
WC2A 3LS.

Telephone No: 0114 275 2400

Patents ADP number — ~~7807043001~~

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number	Country	Priority application number *(if you know it)*	Date of filing *(day/month/year)*
	N/A	N/A	N/A

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application	Number of earlier application	Date of filing *(day/month/year)*
	N/A	N/A

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? — YES

0042190 19-Jul-02 04:01

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	29
Claim(s)	07
Abstract	01 DML
Drawings	15 only

10. If you are also filing any of the following, state how many against each item.

Priority documents	NONE
Translations of priority documents	NONE
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	NONE
Request for preliminary examination and search *(Patents Form 9/77)*	ONE(1)
Request for substantive examination *(Patents Form 10/77)*	NONE
Any other documents *(Please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature

Date Friday, 19 July 2002

12. Name and daytime telephone number of person to contact in the United Kingdom

RALPH ATKINSON CPA
0114 275 2400

SR-07-122

2/77

THE PATENT OFFICE
RM
18 NOV 2003
RECEIVED BY FAX

18NOV03 E848098-1 003028
P07/7700 0.00-0216839.1

Patents Form 7/77

Patents Act 1977

7/77

Statement of inventorship and of right to grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
Gwent NP10 8QQ

1.	Your reference	2034-P585-GB
2.	Patent application number	0216839.1
3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	AUTODESK CANADA INC 10 Duke Street Montreal Quebec Canada H3C 2L7
4.	Title of the invention	PROCESSING SCENE OBJECTS
5.	State how the applicant(s) derived the right from the inventor(s) to be granted a patent	The Applicant derived the right to the invention by virtue of a contract of employment
6.	How many, if any, additional Patents Forms 7/77 are attached to this form? *(see note (c))*	one (1)

7. I/We believe that the person(s) named over the page *(and on any extra copies of this form)* is/are the inventor(s) of the invention which the above patent application relates to.

Signature

Date Tuesday, 18 November 2003

8.	Name and daytime telephone number of person to contact in the United Kingdom	RALPH ATKINSON CPA 0114 275 2400

Patents Form 7/77

0086288 18-Nov-03 01:13

Enter the full names, address and postcodes of the inventors in the boxes and underline the surnames

ERIC <u>PLANTE</u>
4147 rue de Buillone
Montreal
Quebec
Canada H2W 2E6

Patents ADP number: 8754673001

Patents ADP number:

Patents ADP number:

Patents ADP number:

Patents Form 7/77

SR-11-160

Processing Scene Objects

Background of the Invention

1. Field of the Invention

The present invention relates to generating image data as the blending of a plurality of renders of an object along a motion path overtime.

2. Description of the Related Art

Advances in the field of digital signal processing have allowed many new developments to be effected in video and film post production. Many sophisticated image modifications and special effects have become widely accepted as part of post-production procedures.

Often, post-production techniques are used to generate additional image data to be composited within frames of original video or film footage, either because said additional image data cannot be realistically committed to movie or video film, for instance if the movie script requires an actor to jump over a mile-wide gap, or because it does not exist, for instance if the movie script requires photo-realistic alien spaceships.

The degree of realism conveyed by such image data is traditionally paramount in order to immerse an audience within the narrative, and techniques are known with which to motion-blur the additional image data when such data portrays moving objects or actors. Motion is usually a function of the position of an object changing in each frame of a sequence of frame. Displaying such a sequence at the appropriate speed, for instance 24

frames per second for movie films, provides the audience with the illusion of movement. Motion-blurring techniques are used to enhance this illusion, and are especially relevant when the object in question is moving at a fast pace. Motion blurring traditionally involves specifying a shutter length indicative of the number of frames to use to evaluate the motion of an object thereon, and a number of samples which defines how many snapshots are taken of each of said frames, whereby said samples are subsequently blended and the output of said blending operation is an output frame showing said object with motion-blur.

A problem however exists in motion-blurring techniques according to the known prior art, in that additional objects as described above have to be independently motion-blurred if artistic considerations or the movie script requires discrete degrees of motion-blurring for each additional object in the same frame, for instance if two alien spaceships are flying at different speeds, because said motion-blurring techniques according to the known prior art require the shutter length and the number of samples be configured for the entire scene or output frame. This may generate visible artefacts which are highly undesirable to convey the required degree of realism.

Moreover, the independent motion-blurring of said additional objects is very resource-intensive, because a separate output frame may be required for each of said motion-blurred objects, to be composited at a later stage with the original movie or video footage.

A need therefore exists for an improved motion-blurring technique, which is less resource-intensive and allows for generating any additional

objects in a frame at individual degrees of motion-blurring.

Brief Summary of the Invention

According to a first aspect of the present invention, there is provided an apparatus for generating image data comprising memory means configured to store said image data as a volume comprising at least one object and at least one viewport, memory means configured to store motion data for said object, configuration data for said viewport and instructions, and processing means configured by said instructions to perform the steps of defining a motion path for said object within said volume in reply to user input; translating said object along said path over a user-specified period of time; rendering said object through said viewport at portions of said user-specified time period; and blending the resulting plurality of rendered objects to generate said image data.

According to another aspect of the present invention, there is provided a method of generating image data defined as a volume comprising at least one object and at least one viewport, wherein said method includes motion data for said object and configuration data for said viewport, and said method comprises the steps of defining a motion path for said object within said volume in reply to user input; translating said object along said path over a user-specified period of time; rendering said object through said viewport at portions of said user-specified time period; and blending the resulting plurality of rendered objects to generate said image data.

Brief Description of the Several Views of the Drawings

Figure 1 shows an image processing system operated by an image editor;

Figure 2 details the hardware components of the image processing system of *Figure 1* in further detail;

Figure 3 shows motion blur processing to generate image frames according to the known prior art;

Figure 4 details the operational steps according to which a user operates the image processing system of *Figures 1* and *2* according to the invention;

Figure 5 shows the contents of the main memory shown in *Figure 2* after the step of loading instructions and data shown in *Figure 4*;

Figure 6 provides an example of a scene structure including scene data as shown in *Figure 5*;

Figure 7 further details the image data shown in *Figures 5* and *6* as a plurality of objects, including a viewport, within a three-dimensional volume;

Figure 8 illustrates the graphical user interface of the application shown in *Figures 4* and *5*, including a graphical representation of the scene structure and scene data shown in *Figures 6* and *7*;

Figure 9 details the processing steps involved for editing scene data shown in *Figures 4* to *7* as scene objects shown in *Figure 8*;

Figure 10 graphically illustrates the step of equipping an object in the scene of *Figures 6* to *8* with a motion path as shown in *Figure 9*;

Figure 11 details the processing steps according to which image data

is rendered as shown in *Figure 4*;

Figure 12 further details the processing steps according to which samples are processed to generate motion blur shown in *Figure 11*;

Figure 13 shows the objects shown in *Figure 10* sampled and stacked in the memory as shown in *Figure 12*;

Figure 14 details the processing steps according to which the samples shown in *Figure 13* are processed to generate output frame image data;

Figure 15 graphically illustrates the rendering of the objects shown in *Figures 7, 8, 10* and *13* and subsequent blending thereof according to the present invention to generate a complete scene.

Written Description of the Best Mode for Carrying Out the Invention

Figure 1

An image processing system such as a post-production station is illustrated in *Figure 1*. An image editor **101** controls an image processing environment formed by a processing system **102**, a video monitor **103** and a RAID **104**, by means of a keyboard **105**, and a stylus-operated graphics tablet or a mouse **106**. The processing system **102**, such as an Octane™ produced by Silicon Graphics Inc., supplies image signals to the video display unit **103**. Moving image data is stored on the redundant array of inexpensive discs (RAID) **104**. The RAID is configured in such a way as to store a large volume of data, and to supply this data at a high bandwidth, when required, to the processing system **102**. The processing system shown *in Figure 1* is optimal for the purpose of processing image and other

high bandwidth data. In such a system, the instructions for controlling the processing system are complex. The invention relates to any computer system where processing instructions are of significant complexity.

Instructions controlling the processing system **102** may be installed from a physical medium such as a CD-ROM or DVD-ROM **107**, or over a network **108** from a network server **109**, including the Internet **110** accessed therefrom. These instructions enable the processing system **102** to interpret user commands from the keyboard **105** and the mouse or graphics tablet **106**, such that image data, and other data, may be viewed, edited and processed.

Figure 2

The processing system **102** shown in *Figure 1* is detailed in *Figure 2*. The processing system comprises two central processing units **201** and **202** operating in parallel. Each of these processors is a MIPS R11000 manufactured by MIPS Technologies Incorporated, of Mountain View, California. Each of these processors **201** and **202** has a dedicated secondary cache memory **203** and **204** that facilitate per-CPU storage of frequently used instructions and data. Each CPU **201** and **202** further includes separate primary instruction and data cache memory circuits on the same chip, thereby facilitating a further level of processing improvement. A memory controller **205** provides a common connection between the processors **201** and **202** and a main memory **206**. The main memory **206** comprises two gigabytes of dynamic RAM.

The memory controller **205** further facilitates connectivity between the aforementioned components of the processing system **102** and a high bandwidth non-blocking crossbar switch **207**. The switch makes it possible to provide a direct high capacity connection between any of several attached circuits. These include a graphics card **208**. The graphics card **208** generally receives instructions from the processors **201** and **202** to perform various types of graphical image rendering processes, resulting in images, clips and scenes being rendered in real time on the monitor **103**. A high bandwidth SCSI bridge **209** provides an interface to the RAID **104**, and also, optionally, to a digital tape device, for use as backup.

A second SCSI bridge **210** facilitates connection between the crossbar switch **207** and a DVD/CDROM drive **211**. The DVD drive provides a convenient way of receiving large quantities of instructions and data, and is typically used to install instructions for the processing system **102** onto a hard disk drive **212**. Once installed, instructions located on the hard disk drive **212** may be fetched into main memory **206** and then executed by the processors **201** and **202**. An input output (I/O) bridge **213** provides an interface for the mouse or graphics tablet **106** and the keyboard **105**, through which the user is able to provide instructions to the processing system **102**.

Figure 3

Techniques are known to generate motion blur for an object or talent to be composited in a clip of frames at a later stage, in order to convincingly

portray the illusion of movement at speed of said object or talent in the final composited clip. Generating motion blur for an imaginary object according to the known prior art is shown in *Figure 3*.

In order to generate a clip portraying a spaceship **301** travelling at speed, said spaceship **301** is first modelled in a three-dimensional volume **302** with vertices, which eventually define polygons to which textures and a variety of other characteristics may be applied. The volume, or scene **302**, is preferably equipped with a viewport **303**, the purpose of which is to define a view frustrum, the origin of which functions as a camera to render a two-dimensional image of the three-dimensional volume **302** and object **301** therein as seen therethrough.

In order to generate the aforementioned motion blur for spaceship **301** according to the known prior art, it is necessary to render a plurality of images frames **304**, **305**, wherein either spaceship **301** is manually translated within volume **302** after image frame **304** is rendered such that image frame **305** portrays spaceship **301** at a different location, or the position and/or orientation of viewport **303** is altered after image frame **304** is rendered so as to similarly obtain an image frame **305** within which spaceship **301** has moved.

Upon obtaining the two distinct image frames **304**, **305**, a composite image frame **306** portraying spaceship **301** with motion blur **307** is obtained by sampling each of said frames **304**, **305** a number of times. The number of frames sampled which, in this example is two, is known to those skilled in the art as the shutter length, or the sample window size. In the example

according to the prior art described above, five samples **308** to **312** are taken with a shutter length of two frames **304**, **305**.

The samples **308** to **312** are subsequently blended, whereby the two first samples **308**, **309** of image frame **304** carry less weight in the five sample average than the three samples **310** to **312** of image frame **305**, such that the latter position of spaceship **301** within image frame **305** is better defined in composited image frame **306** than the previous position of said spaceship **301** within image frame **304**, shown as a blur **307**.

Figure 4

The processing steps according to which the image processing system **102** of *Figure 1* generates image data according to the invention are further detailed in *Figure 4*.

At step **401**, the image processing system **102** is switched on. At step **402**, the instructions according to the present invention and the data said instructions configure CPUs **201**, **202** to process are loaded from hard disk drive **212**, DVD ROM **107**, network server **109** or the internet **110**, such that said CPUs **201**, **202** may start processing said instructions and data at step **403**.

At step **404**, a scene is selected which, according to the present invention, comprises a structure defined as a hierarchy of data processing nodes and a plurality of types of data to be processed therewith.

The processing of said scene data according to said scene structure generates at least one object, for instance spaceship **301**, within a scene

comparable to three-dimensional volume **302** configured with at least one viewport comparable to viewport **303**, whereby the results of the editing of any of the data defining said object **301**, scene **302** or viewport **303** may be rendered as a frame or a clip of frames at step **406**.

At step **407**, a question is asked as to whether the scene data of another scene should be edited for subsequent rendering. If the question asked at step **407** is answered positively, control is returned to step **404**, whereby the editor **101** may select a different scene structure.

Alternatively, the scene data rendered at step **406** as edited at step **405** is acceptable for the intended purpose of editor **101**, whereby the processing of the instructions according to the invention started at step **403** may now be stopped at step **408** and, eventually, the image processing system **102** switched on at step **401** may eventually be switched off at step **409**.

Figure 5

The contents of the main memory **206** subsequently to the instructions and data loading of step **402** are further detailed in *Figure 5*.

An operating system is shown at **501**, which comprises a reduced set of instructions for CPUs **201** and **202**, the purpose of which is to provide image processing system **102** with basic functionality. Examples of basic functions include access to and management of files stored on hard disk drive **212**, or DVD/CD ROM drive **211**, network connectivity with frame store **104**, server **109** and the internet **110**, interpretation and processing of

the input from keyboard **105** and graphic tablet or mouse **106** and graphical data or binary data output. In the example, the operating system is IRIX™ provided by Silicone Graphics Inc, but it will be apparent to those skilled in the art that the instructions according to the present invention may be easily adapted to function with different other known operating systems, such as Windows™ provided by the Microsoft Corporation of Redmond, California or LINUX which is freely distributed.

An application is shown at **502** which comprises the instructions loaded at step **402**, which enables the image processing system **102** to perform processing steps **404** to **407** according to the present invention within a specific graphical user interface displayed on VDU **103**. A scene structure is shown at **503** and scene data is shown at **504**, which comprises various sets of user input- dependent data and user input-independent data according to which the application shown at **502** generates image data.

Scene structure **503** comprises a plurality of node types **505**, each of which provides a specific functionality in the overall task of rendering a scene according to step **406**. Said node types **505** are structured according to a hierarchy **506**, which may preferably but not necessarily take the form of a database, the purpose of which is to reference the order in which various node types **505** process scene data **504**. Scene structure **503** also includes at least one motion blur configuration file **507** according to the invention, the purpose and functionality of which will be further described hereinafter.

A number of examples of scene data **504** are provided for illustrative

purposes only and it will be readily apparent to those skilled in the art that the subset described is here limited only for the purpose of clarity. Said scene data **504** may include image frames **508** acquired from framestore **104**, audio files **509** such as musical score or voice acting for the scene structure selected at step **404**. Said scene data **504** may also include pre-designed three-dimensional models **510**, such as spaceship **301**, and a variety of textures **511** to apply to said models **510**. In the example, scene data **504** includes lightmaps **512**, the purpose of which is to reduce the computational overhead of CPUs **201**, **202** when rendering the scene with artificial light sources. Scene data **504** finally include three-dimensional location references **513**, the purpose of which is to reference the position of the scene objects edited at step **405** within the three-dimensional volume of the scene.

Figure 6

A simplified example of a scene, or process tree, is shown in *Figure 6* as the scene structure **503** and scene data **504** loaded into memory **206** at step **402**.

Process trees generally consist of sequentially-linked processing nodes, each of which specifies a particular processing task required in order to eventually achieve an output **601**, under the form of a composited frame or a sequence of composited frames. Traditionally, an output scene **601** will comprise both image data and audio data. Accordingly, the composited scene will thus require the output from an image-keying node

602 and the output of a sound mixing node **603**. The image-keying node **602** calls on a plurality of further processing nodes to obtain all of the input data it requires to generate the desired image data. In the example, the desired output image data includes a plurality of frames within which a three-dimensional computer-generated spaceship object **301** is composited within a background consisting of a clip of frames **508** portraying a ship at sea.

The image-keying node **602** therefore initially requires a viewport rendering node **604** to define a frustrum and characteristics thereof within the three-dimensional scene, through which a two-dimensional rendering of three-dimensional objects within a scene may take place. The image-keying node **602** subsequently requires the sequence of frames **508** from frame node **605**, each frame of which is processed by a colour-correction processing node **606** and a motion tracking processing node **607**, such that the composited three-dimensional spaceship object **301** generated by three-dimensional modelling node **608**, to which is applied a texture **511** by the texturing node **609** and appropriate lightmaps **512** by processing node **610** and which is also motion-tracked by processing node **607**, is seamlessly composited within the colour corrected sequence of frames **508**.

In so far as the lighting of spaceship **301** is concerned, the image keying processing node **602** also requires the output of a spotlight generated by an artificial light processing node **611** within the scene to interact with the lightmaps **512** of spaceship **301**, wherein said artificial light

is colour-corrected by a colour-correction processing node **612** providing a functionality similar to the functionality of colour-correction processing node **606**. A filter object is preferably generated by a light filtering processing node **613** to prevent the artificial light of said spotlight from altering the colour characteristics of the frames **508** within the scene.

In the preferred embodiment of the present invention, all of the data generated by the above described nodes may be visualised as distinct three-dimensional objects within a scene defined as a three-dimensional volume configured with Cartesian x, y and z co-ordinates, whereby motion tracking processing node **607** processes the x, y and z co-ordinates of each of said objects. The image-keying processing node **602** subsequently overlays said three-dimensional objects as viewed through the frustrum generated by node **604**.

Figure 7

The scene data **504** generated as three-dimensional objects by scene structure **503** described in *Figure 6* are shown within a scene defined as a three-dimensional volume in *Figure 7*.

The textured and lightmap-configured spaceship model **301** is shown within scene **302** in relation to the viewport **303** generated by processing node **604**. Said viewport is configured with a view frustrum **701** and a focal length **702**, which jointly define the two-dimensional plane **703** corresponding to the required image output data generated by output processing node **601**. Said two-dimensional plane **703** may simply be

thought of as the image frame that would be committed to film, were viewport **303** a conventional camera filming three-dimensional objects within scene **302**. Said two-dimensional plane **703** will thus be hereinafter referred to as a rendering window.

The clip of frames **508** generated by node **605** is shown as a two-dimensional plane **704** equipped with x, y and z co-ordinate within volume **302**, wherein the area of said plane **704** is defined by the resolution in pixels or lines of the image frames **508**. Said plane **704** is known to those skilled in the art as a billboard and, depending upon whether the functionality of window **704** allows the entire clip of frames **508** to be played therein, may also be known as a player.

In the example, volume **302** also includes a spotlight object **705** generated by artificial light node **611** and a filter **706** generated by node **613**. Said filter **706** is preferably positioned by motion-tracking node **607** between spotlight **705** and player **704** in order to refract artificial light cast by said spotlight **705** within scene **302** to light spaceship **301** and thus prevent said artificial light from distorting the colour component values of frames **508** within player **704**.

In the example still, the clip of image frames **508** portrays a ship **707** at sea, wherein said image frames were shot with a camera aboard a different ship. Consequently, the combined motions of the ship **707** and the camera aboard said other ship arising from random sea surface movements result in the level **708** of the sea to alter substantially over the entire duration of the clip of frames **508**, whereby ship **707** in effect rises

and falls along a vertical path within player **704**.

Image editor **101** wishes to remove this vertical motion in the composited output clip, effectively tracking a fixed portion of the frame in each frame of the clip of frames **508**. However, image editor **101** also wishes to composite spaceship **301** as moving at speed towards a position **709** over said ship **707**, which is stationary, in each of said frames **508**. A problem therefore exists in that spaceship **301** requires a degree of motion blur to realistically convey the impression of said movement at speed, whilst the ship **707** requires no motion blurring at all, because it is stationary.

With reference to the known prior art described in *Figure 3*, conventional compositing techniques would require image processing system **102** to generate a first clip of frames portraying spaceship **301** moving towards position **709** with motion blurring, the total number of frames of which equals the total number of frames of the clip of frames **508**. A second output clip should then be generated as portraying stationary ship **707** with a stable sea level **708**, for instance with processing the clip of frames **508** with the "stabiliser with roll" disclosed in United States patent No 5,786,824 assigned to the present applicant. The required result would be achieved by keying the first output clip portraying spaceship **301** moving at speed and the second output clip portraying a stabilised, stationary ship **707**.

According to the present invention, however, the above required output clip is generated by means of defining respective motion paths for both spaceship **301** and player **704** within scene **302**, respectively

translating said spaceship **301** and player **704** along said paths over a period of time equivalent to the duration of clip frames **508**, rendering spaceship **301** and player **704** at portions of said period of time, which is also known to those skilled in the art as sampling, and blending the resulting plurality of samples in order to generate an output image frame or an output clip of image frames.

Figure 8

The graphical user interface (GUI) of the application **502** is shown in *Figure 8*, including a graphical representation of the scene structure **503** and scene data **504** shown in *Figures 5* and *6* and further described in *Figure 7*.

The GUI **801** of image processing application **502** is preferably divided into a plurality of functional areas, most of which are user-operable. A first area **802** displays scene structure **503** as three-dimensional scene **302**. Said scene **302** is preferably displayed with including scene data **504** graphically depicted as scene objects **301**, **303** and **701** to **708**.

A cursor **803** is displayed which is user operable by means of mouse or graphic tablet **106** and may be positioned by image editor **101** over any portion of GUI **801** to select a variety of functions or tasks within said plurality of functional areas. Thus, within scene display area **802**, cursor **803** may be used to select a particular object, for instance spaceship **301**, whereby a second user operable area **804** conveys data properties, parameters and/or values specifically pertaining to said selected object.

Preferably, second object information display area **804** includes an object identifier portion **805**, an object location portion **806** and an objet properties portion **807**. Portions **805** and **806** of second display area **804** are updated according to which object cursor **803** selects within scene **302** and portion **807** may be subsequently interacted therewith by means of said cursor **803** to edit any of the selected object's properties. .

A third display area **808** comprises conventional user-operable clip navigation widgets allowing image editor **101** to respectively rewind, reverse play, pause, stop, play or fast forward the sequential order of image frames generated from scene **302** by means of rendering window **703**. Alternatively, said navigation widgets **808** also provide the same functionality as described above for player **704** if said player is selected as a scene object by means of cursor **803**. A counter area **809** is provided in close proximity to the clip navigation widget **808**, which is divided into hours, minutes, seconds and frames, such that the aforementioned navigation by means of navigation widgets **808** may be carried out with precision and provide a valid point of reference to image editor **101**.

A fourth display area **810** provides a conventional bar of menus operable by means of cursor **803**, which provide a variety of functions and processes, for instance with which to load or store image data, further configure the size and contents of display areas **802**, **804** and **808** or, eventually, stop processing the instructions according to the invention according to step **409**.

Figure 9

The processing step **405** according to which scene data **504** is edited as scene objects shown in *Figures 7* and *8* is further described in *Figure 9*.

At step **901**, the required output clip length is set. Said length may be understood as the number of output image frames node **602** should render through rendering window **703** in order to generate a complete sequence of frames defining a clip and, in the example, said output clip length is equivalent to either the number of frames **508** of the clip shown as player **704** or, alternatively, the number of frames required to portray spaceship **301** moving at speed to position **709** within scene **302**.

Upon specifying the number of frames to output, a first object such as spaceship **301** is selected in the scene **302**, for instance by means of cursor **803**, at step **902**. A path may subsequently be defined for said selected object within said scene **302** at step **903**, for instance by selecting the "path" property of said object within object's properties portion **807**.

In the preferred embodiment of the present invention, said path is linear and comprises a directional vector, the origin which is defined by the x, y and z co-ordinates of the object selected at step **902** and the extremity of which is defined by the x, y and z co-ordinates of said object subsequently to image editor **101** dragging said object **301** to position **709**, for instance with using cursor **803** in a "click and drag" configuration, which is well known to those skilled in the art. In an alternative embodiment of the present invention, however, said path comprises a spline-based curve, for

instance to impart a "bobbing" motion to spaceship **301**. In yet another alternative embodiment of the present invention, said path is defined as a function, for instance the tracking function disclosed in United States Patent No. 5,786,824 referenced thereabove, wherein said function itself generates a linear or spline-based movement vector.

The path configuration input at step **903** is subsequently stored at step **904** as three-dimensional locations **510**, specifically for the object selected at step **902**. At step **905**, the shutter length specifically relating to the selected object is input. A question is subsequently asked at step **906**, as to whether a path should be defined for another object in scene **302**. If the question asked at step **906** is answered positively, control is returned to step **902**, whereby said next object may be selected and a respective path defined and stored. Alternatively, if the question of step **906** is answered negatively, a sample number is input for the scene as a whole.

According to the known prior art, and with reference to *Figure 3*, shutter length and the number of samples are traditionally specified for the entire scene. Specifying the shutter length and number of samples for the entire scene **302** as described in *Figures 7* and *8* according to said prior art would result in generating appropriate motion blurring for spaceship **301**, but would also generate motion blurring for ship **707** which, in the example, is undesirable. According to the present invention, shutter length is input for each object in scene **302** independently of the total number of samples required for the scene, whereby it is thus possible to generate a two-dimensional image frame within which each of said plurality of objects is

rendered with its own degree of motion blurring such that, in the example, the motion blurring of ship **707** within the original clip of frames **508** is removed whilst appropriate motion blurring is convincingly provided for spaceship **301**.

Figure 10

The processing step **903** of equipping an object in scene **302** of *Figures 7* and *8* with a motion path is graphically illustrated in *Figure 10*.

With reference to the description of step **903**, image editor **101** first selects spaceship object **301** with cursor **803** and drags it within scene **302** to location **709**, whereby a linear directional vector **1001** is defined with an origin, the x, y and z co-ordinates of which are obtained from the location of said object **301** before interaction therewith, and an extremity **709**, the x, y and z co-ordinates of which within scene **302** are derived from the location to which the cursor **803** releases spaceship **301** after dragging.

Preferably, spaceship **301** is selected according to step **902**, whereby cursor **803** subsequently selects the path property of said object within portion **807** such that application **502** initiates the vector origin and continually reads the input data of mouse or graphic tablet **106**, e.g. cursor **803**, to define the length and orientation of said vector within scene **302**.

Similarly, user **101** selects the player **704** as a next object in scene **302** according to step **902**, again selecting the path property of said player within updated portion **807** but specifies said path as a tracking function as opposed to a directional vector. Image editor **101** may subsequently select

a two-dimensional x, y portion **1002** of the image frame area within player **704**, whereby said portion will be tracked in each subsequent frame in said frame display area of said player **704** by means of calculating a movement vector, the inverse of which will be applied to player object **704** such that the x, y portion **1002** remains stable in relation to the two-dimensional rendering window **703**.

According to the present invention, individual motion paths may therefore be defined for a number **n** of objects representing scene data within scene **302**, whereby each of said distinct motion paths may be linear, non-linear or the result of an object processing function, in relation to said rendering window **703**.

Figure 11

The processing steps according to which image data edited as scene objects is rendered at rendering step **406** are further detailed in *Figure 11*.

At step **1101**, a frame counter is initialised with the number of frames defined by the output clip length configured at step **901** and the first frame of the output clip or output image frame is selected for rendering.

At step **1102**, the number of samples input at step **907** are processed in order to generate the image data required to create or remove the respective degree of motion blurring for each of the objects in scene **302** to be rendered as viewed through rendering window **703**. Upon generating the required image data as samples of picture screen element,

also known as pixels, each of which having Red (R), Green (G) and Blue (B) colour component values, said samples are processed in order to generate the target pixels defining the output image data, i.e. the output frame, at step **1103**.

At step **1104**, the output frame generated at step **1103** is subtracted from the number of frames initialised in the counter at step **1101**, whereby a question is asked as to whether all frames of the required output clip have been rendered. If the question of **1104** is answered negatively, control is returned to step **1101**, whereby the next output frame is selected for rendering according to steps **1102** and **1103** and the counter further decreased at step **1104** until such time as question **1104** is answered positively, whereby all of the required output image data has been rendered.

Figure 12

The processing steps according to which samples are processed at step **1102** in order to generate image data to create motion blur are further detailed in *Figure 12*.

At step **1201**, application **502** reads the motion blur configuration files **507** of each object in scene **302**, to derive the maximum shutter length input for anyone of said objects and set said maximum shutter length as the total number of "unblurred" frames to render for sampling and subsequent processing in order to generate the output frame selected at step **1101**. At step **1202**, the first frame of said total number of "unblurred" frames is

rendered and sampled, e.g. the RGB colour component values of each pixel of said frame are read by application **502** such that they may be stacked in memory **206** at the next step **1203**.

At step **1204**, a question is asked as to whether another "unblurred" frame should be sampled and, if said question is answered affirmatively, the respective x, y and z co-ordinates of each object within scene **302** are interpolated along the respective motion paths of said objects at step **1205**, such that the next frame to be sampled may be rendered with the scene objects at interpolated positions at step **1206**. Control is subsequently returned to step **1202**, whereby said next rendered frame is sampled and said sample is stacked in memory **206** at step **1203**.

The question of step **1204** is eventually answered negatively, signifying that all of the required samples of "unblurred" frames have been obtained and may now be processed in order to generate the target pixels.

Figure 13

A graphical representation of the sampling and stacking of processing steps **1202**, **1203** as well as the rendering step **1206** is provided in *Figure 13*.

In the example, it was previously explained that ship **707** should have no motion blurring whatsoever whereas spaceship **301** does require motion blurring to convey the impression of movement at speed. Moreover, in the preferred embodiment of the present invention, only a tracked portion of player **704** is required for the final output image data.

Consequently, image editor **101** specifies a shutter length of one frame for player object **704**, whereby regardless of the number of additional "unblurred" frames to be rendered according to the frame total of step **1201**, the x, y and z position of said player object **704** will not be interpolated and thus said object will appear immobile regardless of the number of samples taken and stacked according to steps **1202** and **1203**. Conversely, image editor **101** specifies a shutter length of two frames for spaceship object **301**, such that the number of samples specified for the scene at step **907** is equally divided between said two "unblurred" frames in the case of uniform sampling or, alternatively, more samples are taken of one "unblurred" frame than the other in the case of weighed sampling. Both concepts of uniform sampling and weighed sampling will be familiar to those skilled in the art and, for the purpose of clarity, only uniform sampling will be hereinafter described in the present description.

In the example, six samples should be taken and stacked to generate each output image frame and appropriate input is received according to step **907**, thus three samples of each of the two "unblurred" frames shall be taken and stacked according to steps **1202**, **1203**.

In accordance with the description of the present invention, a first "unblurred" frame **1301** is thus generated, the size i.e. resolution of which is defined by tracked portion **1002** of player object **704** and spaceship object **301**, both of which are respectively at the origin of their respective motion paths **1001**, **1002**. Three sample **1302**, **1303** and **1304** are taken of said first frame **1301**, whereby question **1204** is answered positively and, in

accordance with the above description, only the position of spaceship objects **301** is interpolated to position **1305** along directional vector **1001** to generate a second "unblurred" frame **1306**, three samples **1307**, **1308** and **1309** of which are subsequently taken.

The stacking of the six samples **1302** to **1304** and **1307** to **1309** is figuratively represented at **1310** and may be thought of as the superimposition of said six samples, the sum total of the colour component values of each corresponding pixel of which is divided by the total number of samples to provide a single image frame comprising the target pixels.

Figure 14

The processing steps according to which application **502** processes the samples stacked according to step **1203** to generate output frame image data **1310** are further detailed in *Figure 14*.

At step **1401**, the first target pixel of the output image frame **1310** is selected, the respective red, green and blue colour component values of which equal the average of the respective red, green and blue colour component values of the pixels having the same two-dimensional co-ordinates in all of the samples stacked.

In order to generate said average, each sample is recursively selected at step **1402** such that the pixel therein having x, y co-ordinates equivalent to the x, y co-ordinates of the selected target pixel of step **1401** may be selected at step **1403** and its respective colour component values added at step **1404**. At step **1405**, a first question is asked as to whether all

of the samples have been traversed and processed, thus providing total colour component values which may be subsequently divided by the total number of samples traversed at step **1406** to obtain the final RGB colour component value of the target pixels selected at step **1401**, if the question of step **1405** is answered positively.

Alternatively, if the question of step **1405** is answered negatively, the next sample is selected at step **1402**, traversed to identify the corresponding pixel at step **1403**, the RGB values of which are added according to step **1404**. Having obtained final TGB colour component values for the target pixel currently selected at step **1401**, a second question is asked at step **1407** as to whether all of the target pixels defining the target image frame have been processed, such that the next output image frame may be generated for the final output clip. If the question of step **1407** is answered negatively, control returns to step **1401**, whereby the next pixel of the target image frame is selected and its final RGB colour component values calculated according to step **1402 to 1406**.

Question **1407** is eventually answered positively, whereby an output image frame is generated which includes a plurality of objects, each of which having its own degree of motion blurring.

Figure 15

The rendering of objects in scene **302** at step **406** according to the present invention is shown in *Figure 15* in order to generate a complete scene comprising a clip of a plurality of output image frames.

For the purpose of clarity, only three output image frames **1501**, **1502** and **1503** are shown from an output clip including potentially hundreds or even thousands of image frames generated according to the present invention. Image data defining image frame **1501** is generated according to the processors described in *Figure 13*, wherein motion blurring **1504** is generated by blending image frames **1301** and **1306** according to the processing steps described in *Figure 14*.

The resolution of frame **1501** is derived from the resolution of the tracked portion **1002** of player **704** and includes spaceship **301**. As the sampling and stacking processes of steps **1202** to **1206** are carried out to generate the entire clip, the position of spaceship **301** is regularly interpolated along its directional vector, e.g. motion path **1001**. However, the tracking function invoked to stabilise player portion **1002** eventually generates a directional vector **1505**, the inverse **1506** of which is applied to the entire player object **704** such that ship **707** remains stabilised in output image frame **1502**, whilst object **301** is still rendered with motion blur **1504**.

Similarly, the tracking function eventually generates another, different directional vector **1507**, the inverse of which is yet again applied to the entire player object **704** to achieve the same result in output image frame **1503** as the result described in output image frame **1502**.

A method is therefore provided for generating image data **1501** to **1503** define as a volume **302** comprising at least one object **301** and at least one viewport **303**, wherein a motion path **1001** is defined for said object **301**, the object **301** is translated along said motion path **1001** over a

period of time specified at **901**, said object **301** is rendered (**1301**, **1306**) through said viewport **303** at sampling portions **1302**, **1303**, **1304**, **1307**, **1308** and **1309** and the resulting plurality of renders (**1301**, **1306**) are blended (**1310**) to generate said image data **1501**.

Claims:

1. Apparatus for generating image data comprising memory means configured to store said image data as a volume comprising at least one object and at least one viewport, memory means configured to store motion data for said object, configuration data for said viewport and instructions, and processing means configured by said instructions to perform the steps of

defining said motion data as a motion path for said object within said volume in reply to user input;

translating said object along said path over a user-specified period of time;

rendering said object through said viewport at portions of said user-specified time period; and

blending the resulting plurality of rendered objects to generate said image data.

2. Apparatus according to claim 1, wherein said volume comprises a plurality of objects.

3. Apparatus according to claim 2, wherein motion paths are respectively defined for each of said objects, said objects are translated

and rendered at said portions of said user-specified time period and blended to generate said image data.

4. Apparatus according to any claims 1 to 3, wherein said motion path is a directional vector, a definition of which is any of a group comprising a linear vector, a spline-based vector or a motion data processing function.

5. Apparatus according to claim 4, wherein said motion data processing function is the tracking of a portion of an object.

6. Apparatus according to any of claims 1 to 5, wherein said configuration data for said viewport comprises a view frustrum and a focal length.

7. Apparatus according to any of claims 1 to 6, wherein said translating step further includes the step of interpolating the position of said object along said motion path.

8. Apparatus according to any of claims 1 to 7, wherein said user-specified period of time is a number of frames.

9. Apparatus according to claim any of claims 1 to 8, wherein said portions of said user-specified period of time are samples.

10. Apparatus according to any of claims 1 to 9, wherein said blending step generates motion blurring for said object in said image data.

11. A method for generating image data comprising said image data as a volume including at least one object and at least one viewport, wherein said method comprises the steps of

defining a motion path for said object within said volume in reply to user input;

translating said object along said path over a user-specified period of time;

rendering said object through said viewport at portions of said user-specified time period; and

blending the resulting plurality of rendered objects to generate said image data.

12. A method according to claim 11, wherein said volume comprises a plurality of objects.

13. A method according to claim **12**, wherein motion paths are respectively defined for each of said objects, said objects are translated and rendered at said portions of said user-specified time period and blended to generate said image data.

14. A method according to any claims **11** to **13**, wherein said motion path is a directional vector, a definition of which is any of a group comprising a linear vector, a spline-based vector or a motion data processing function.

15. A method according to claim **14**, wherein said motion data processing function is the tracking of a portion of an object.

16. A method according to any of claims **11** to **15**, wherein said configuration data for said viewport comprises a view frustrum and a focal length.

17. A method according to any of claims **11** to **16**, wherein said translating step further includes the step of interpolating the position of said object along said motion path.

18. A method according to any of claims 11 to 17, wherein said user-specified period of time is a number of frames.

19. A method according to claim any of claims 11 to 18, wherein said portions of said user-specified period of time are samples.

20. A method according to any of claims 11 to 19, wherein said blending step generates motion blurring for said object in said image data.

21. A computer readable medium having computer readable instructions executable by a computer, such that said computer performs the steps of:

defining said image data as a volume including at least one object and at least one viewport;

defining a motion path for said object within said volume in reply to user input;

translating said object along said path over a user-specified period of time;

rendering said object through said viewport at portions of said user-specified time period; and

blending the resulting plurality of rendered objects to generate said

image data.

22. A computer readable medium according to claim **21**, wherein said volume comprises a plurality of objects.

23. A computer readable medium according to claim **22**, wherein motion paths are respectively defined for each of said objects, said objects are translated and rendered at said portions of said user-specified time period and blended to generate said image data.

24. A computer readable medium according to any claims **21** to **23**, wherein said motion path is a directional vector, a definition of which is any of a group comprising a linear vector, a spline-based vector or a motion data processing function.

25. A computer readable medium according to any of claims **21** to **24**, wherein said translating step further includes the step of interpolating the position of said object along said motion path.

26. A computer readable medium according to any of claims **21** to **25**, wherein said blending step generates motion blurring for said object in said image data.

27. A computer system programmed to generate image data, including memory means configured to store said image data as a volume comprising at least one object and at least one viewport, memory means configured to store motion data for said object, configuration data for said viewport and instructions, and processing means configured by said instructions to perform the steps of

defining said motion data as a motion path for said object within said volume in reply to user input;

translating said object along said path over a user-specified period of time;

rendering said object through said viewport at portions of said user-specified time period; and

blending the resulting plurality of rendered objects to generate said image data.

Abstract of the Disclosure

PROCESSING SCENE OBJECTS

A method is provided for generating image data **1310** defined as a volume **302** comprising at least one object **301** and at least one viewport **303**, wherein a motion path **1001** is defined for said object **301**, the object **301** is translated (**1205**) along said motion path **1001** over a period of user-specified time (**901**), said object **301** is rendered (**1301, 1306**) through said viewport **303** at sampling portions (**1302, 1303, 1304, 1307, 1308, 1309**) of said user-specified time and the resulting plurality of renders (**1301, 1306**) are blended (**1310**) to generate said image data **1310**.

(Figure 7)

2034-P585-GB

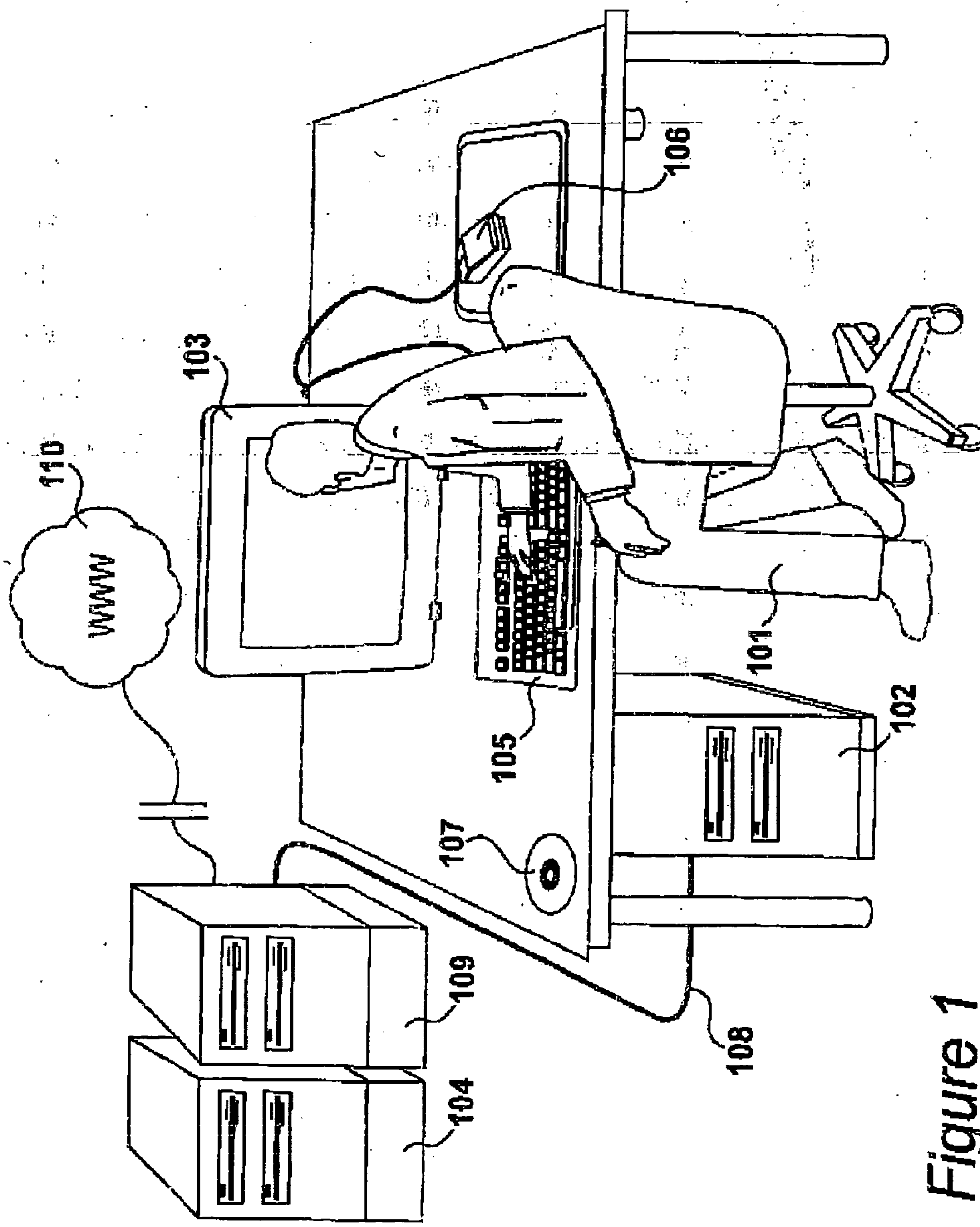


Figure 1

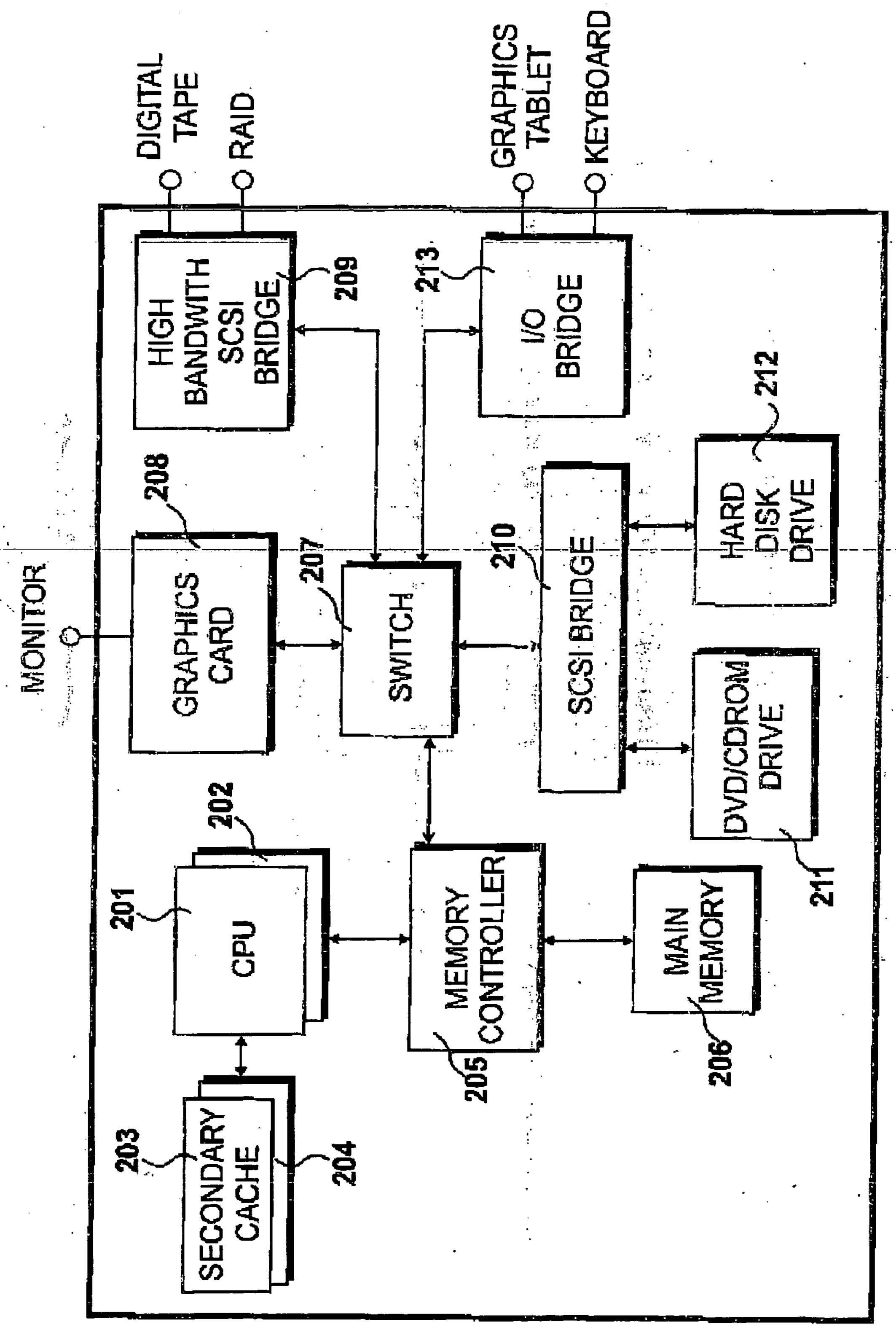

DIGITAL TAPE
RAID
GRAPHICS TABLET
KEYBOARD
HIGH BANDWITH SCSI BRIDGE
209
213
I/O BRIDGE
212
HARD DISK DRIVE
208
MONITOR
GRAPHICS CARD
207
SWITCH
210
SCSI BRIDGE
DVD/CDROM DRIVE
211
202
201
CPU
MEMORY CONTROLLER
205
MAIN MEMORY
206
203
SECONDARY CACHE
204

Figure 2

2034-P585-GB

3/15

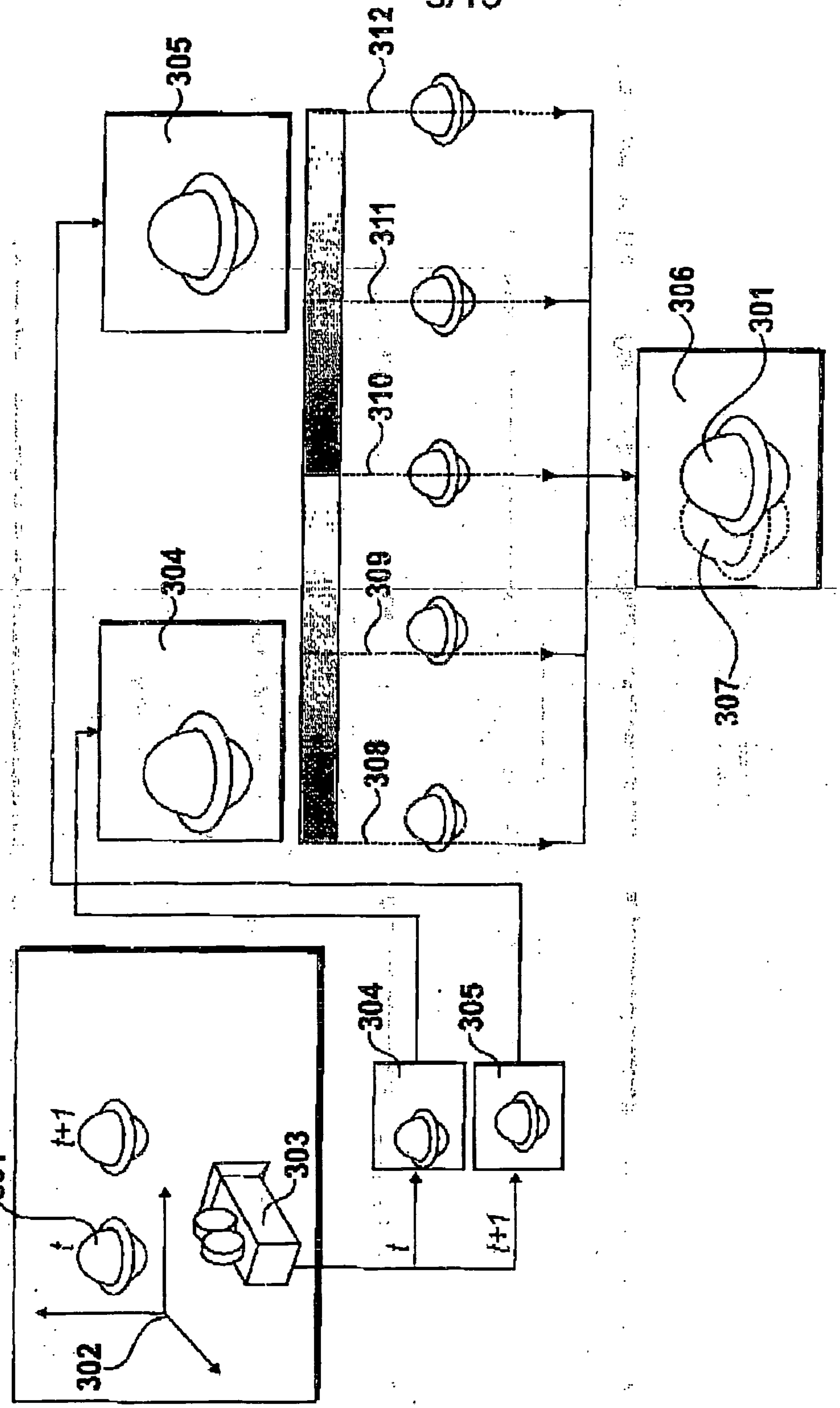
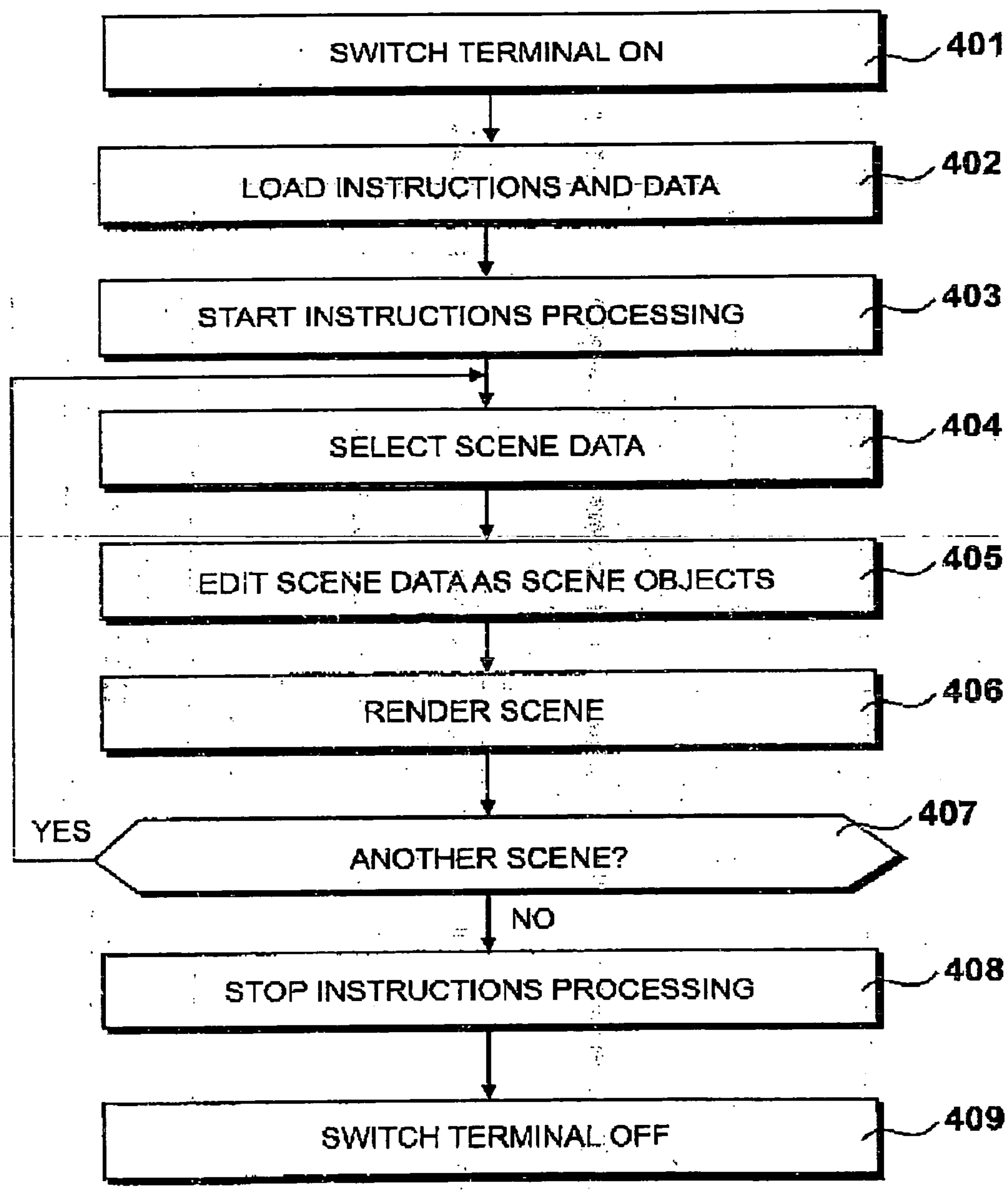


Figure 3

[PRIOR ART]



SWITCH TERMINAL ON
401
LOAD INSTRUCTIONS AND DATA
402
START INSTRUCTIONS PROCESSING
403
SELECT SCENE DATA
404
EDIT SCENE DATA AS SCENE OBJECTS
405
RENDER SCENE
406
ANOTHER SCENE?
407
YES
NO
STOP INSTRUCTIONS PROCESSING
408
SWITCH TERMINAL OFF
409

Figure 4

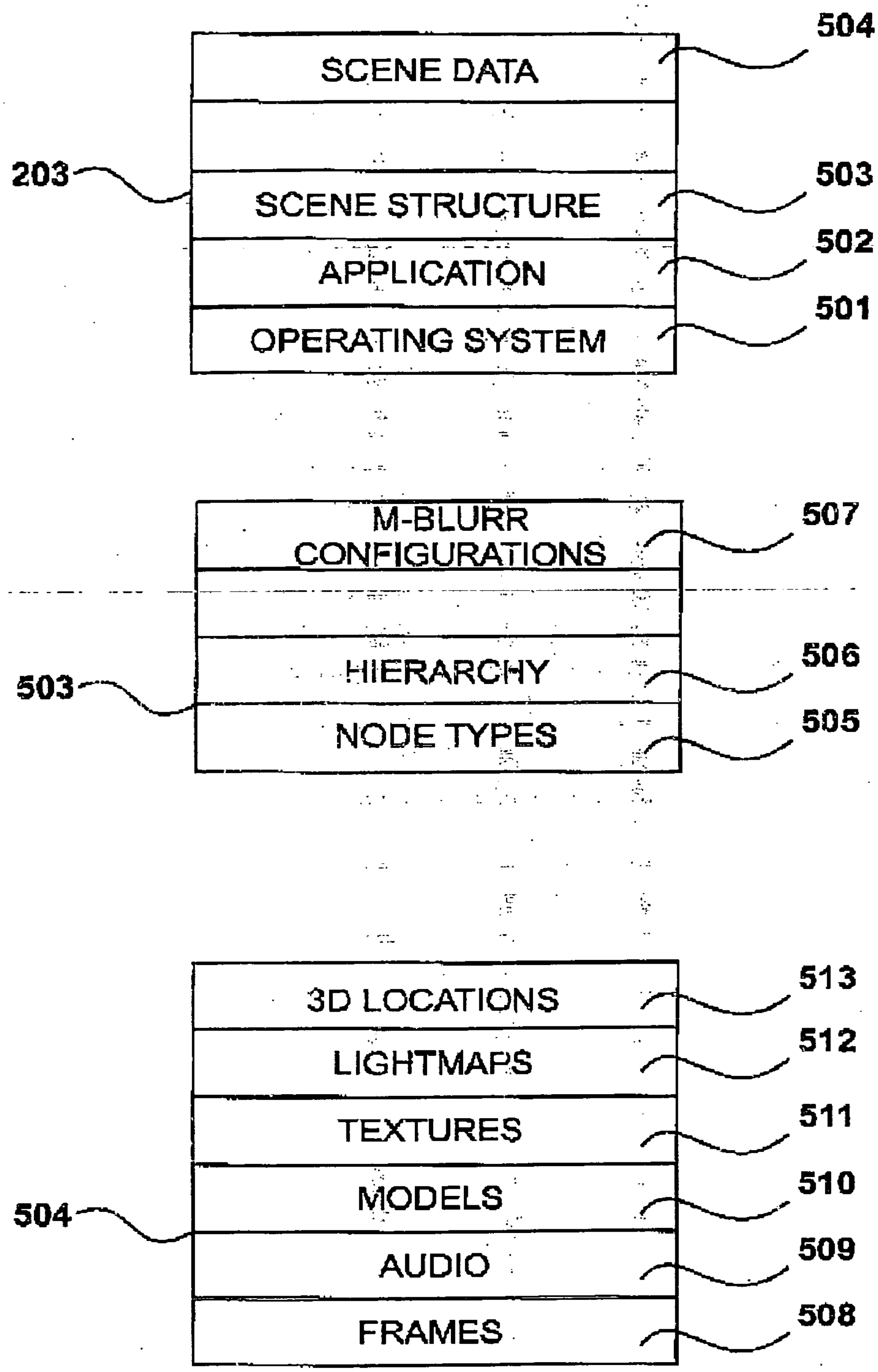

504
SCENE DATA
203
503
SCENE STRUCTURE
502
APPLICATION
501
OPERATING SYSTEM
507
M-BLURR CONFIGURATIONS
506
503
HIERARCHY
505
NODE TYPES
513
3D LOCATIONS
512
LIGHTMAPS
511
TEXTURES
510
504
MODELS
509
AUDIO
508
FRAMES

Figure 5

2034-P585-GB

6/15

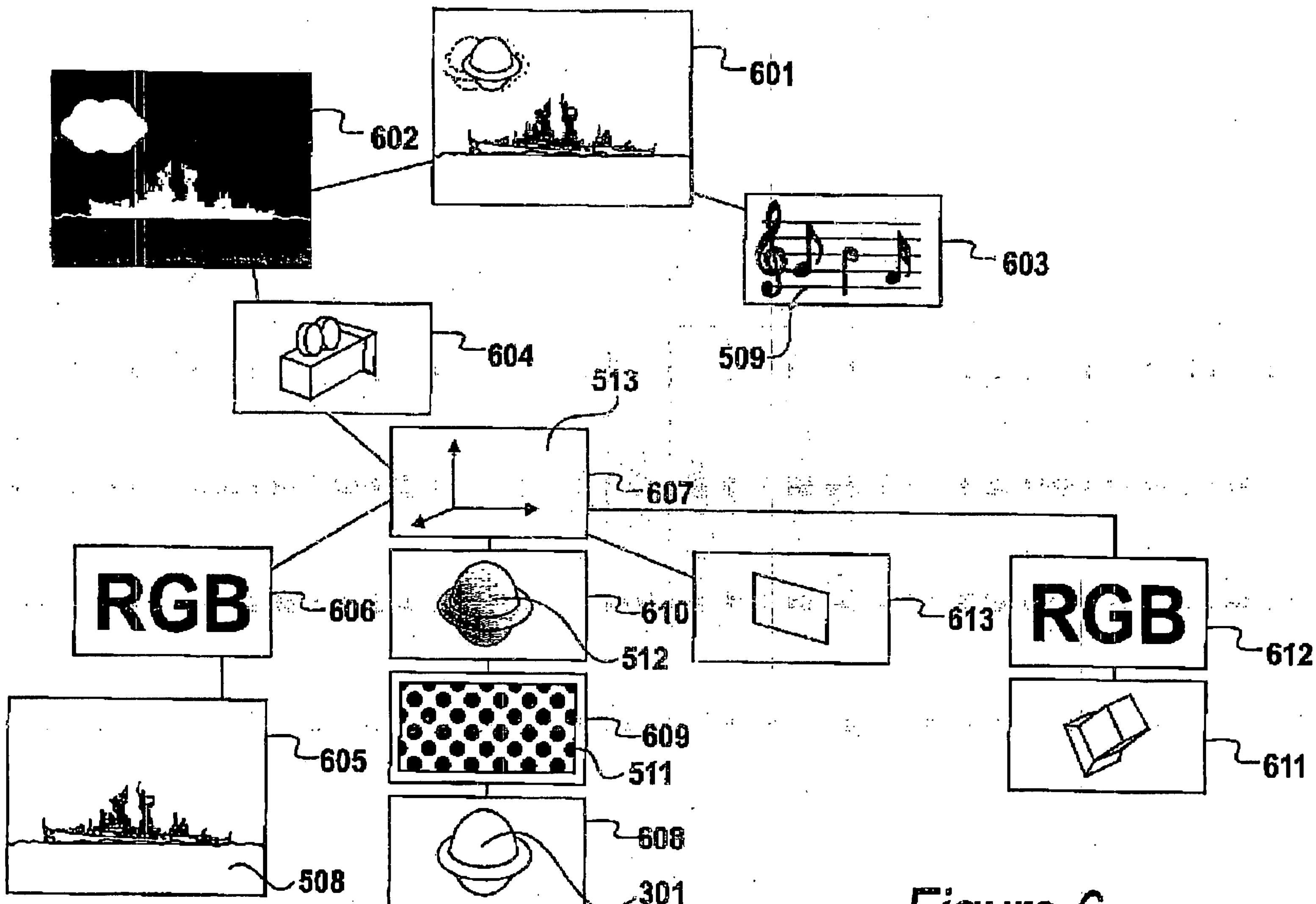


Figure 6

2034-P585-GB

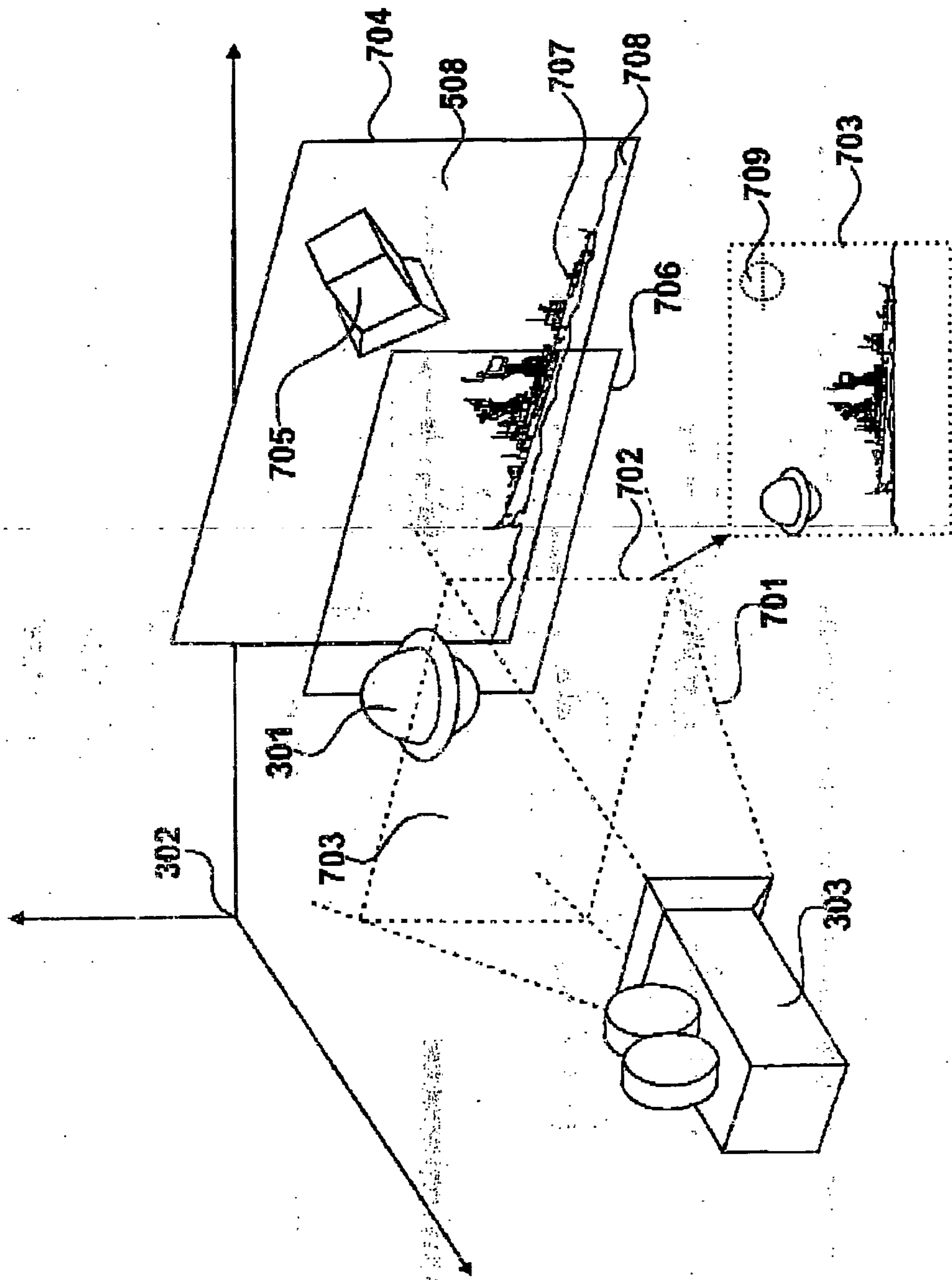


Figure 7

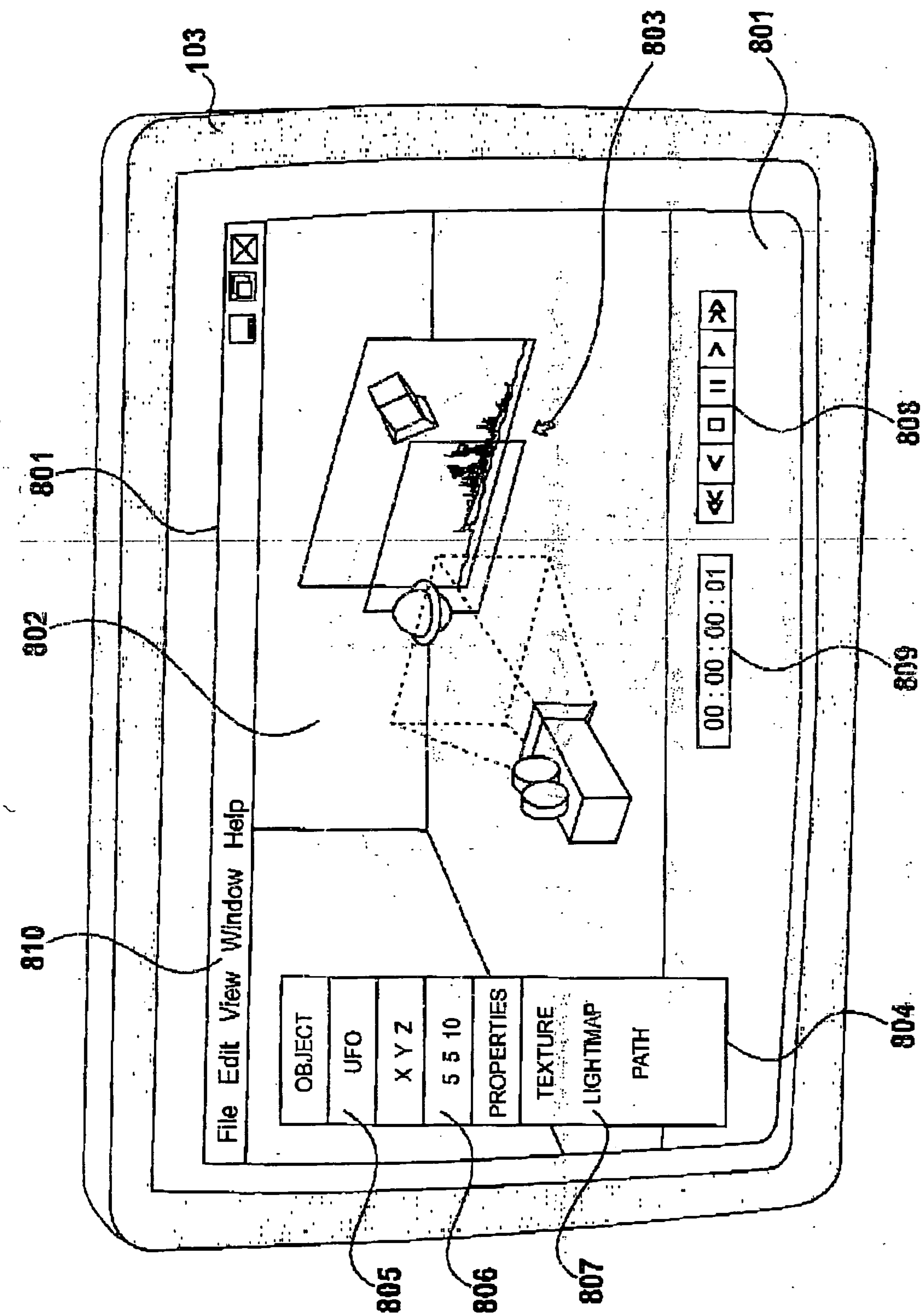

103
803
801
801
802
810
808
809
804
805
806
807
File Edit View Window Help
OBJECT
UFO
X Y Z
5 5 10
PROPERTIES
TEXTURE
LIGHTMAP
PATH
00 : 00 : 00 : 01

Figure 8

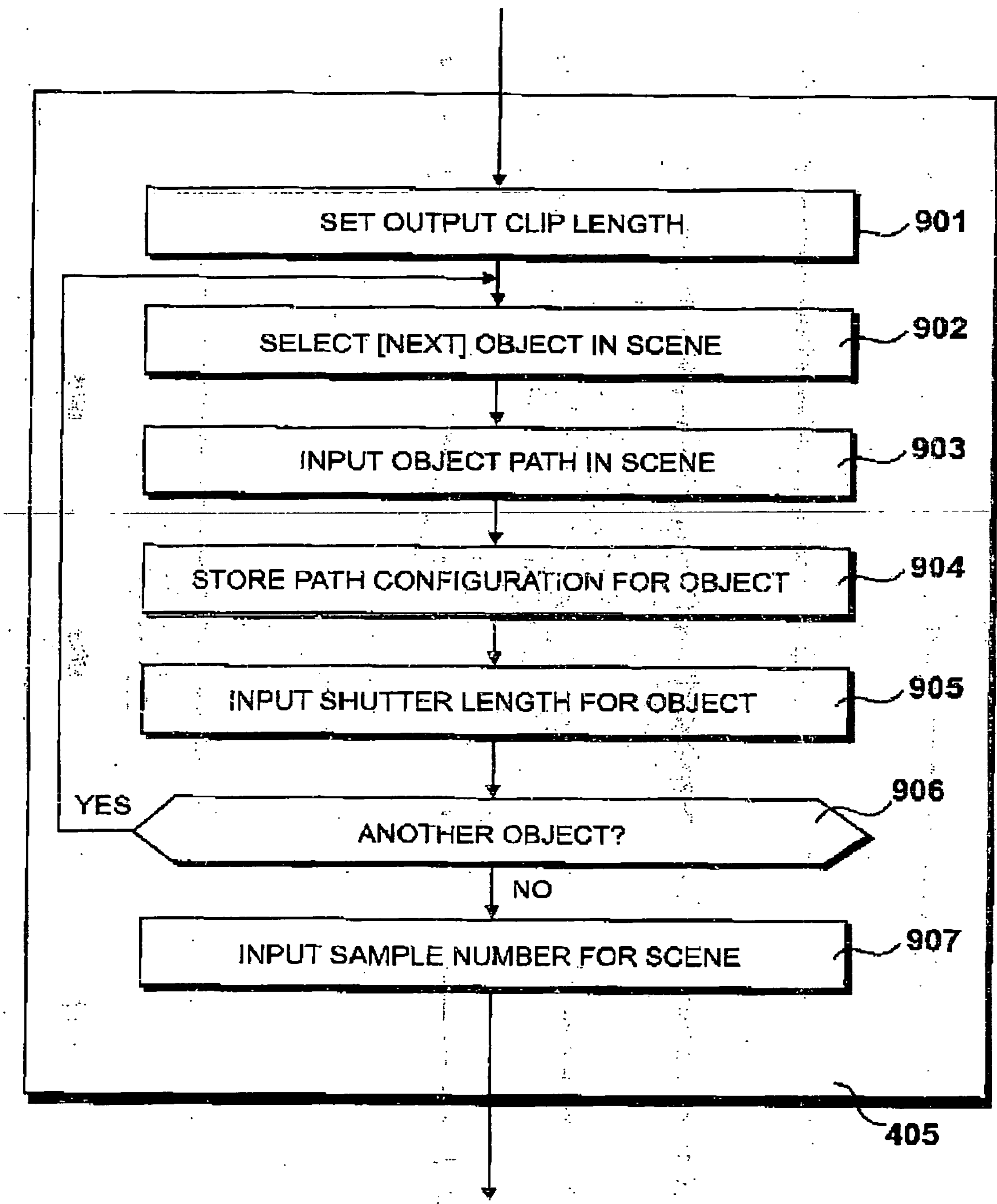

SET OUTPUT CLIP LENGTH
901
SELECT [NEXT] OBJECT IN SCENE
902
INPUT OBJECT PATH IN SCENE
903
STORE PATH CONFIGURATION FOR OBJECT
904
INPUT SHUTTER LENGTH FOR OBJECT
905
YES
ANOTHER OBJECT?
906
NO
INPUT SAMPLE NUMBER FOR SCENE
907
405

Figure 9

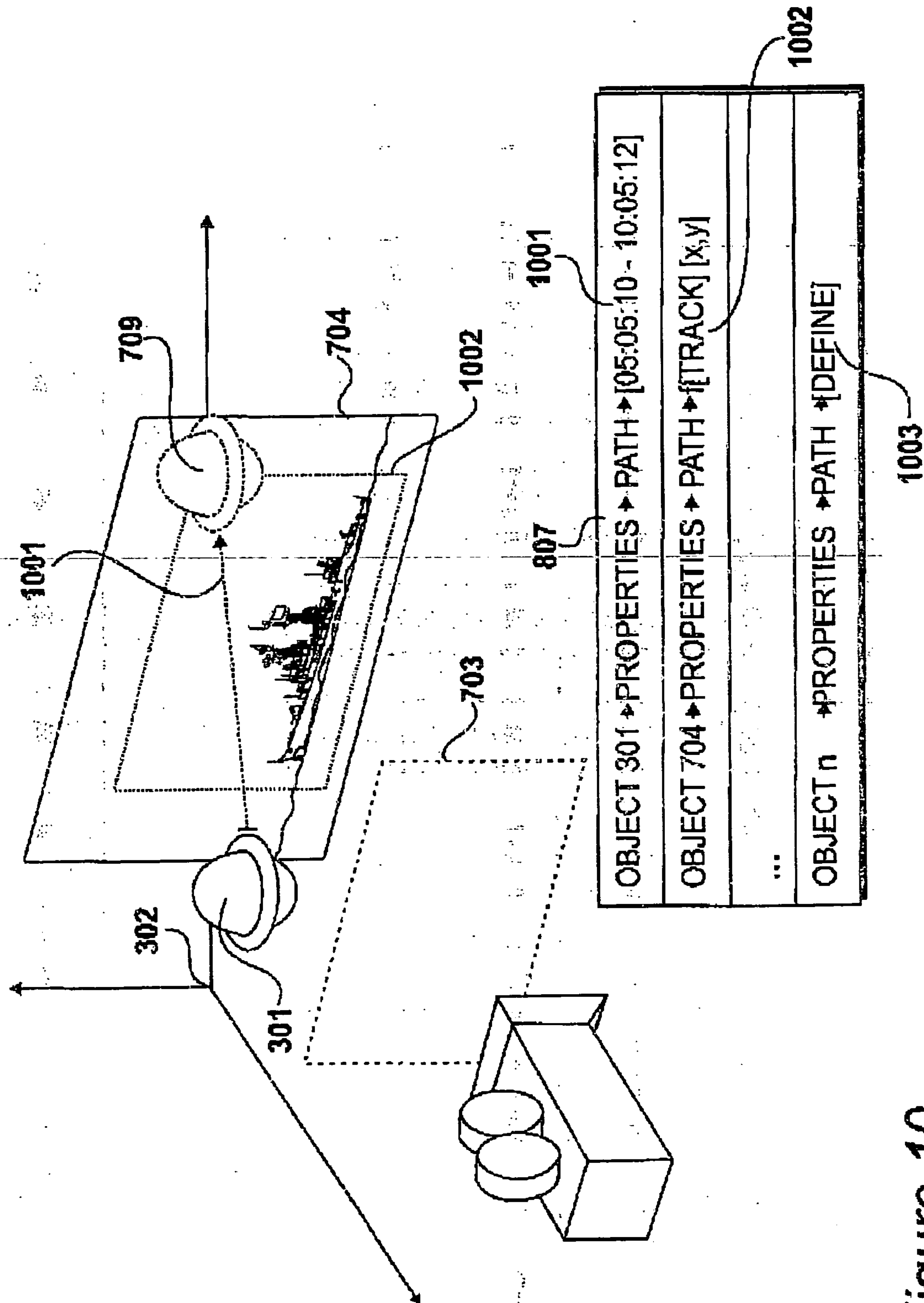
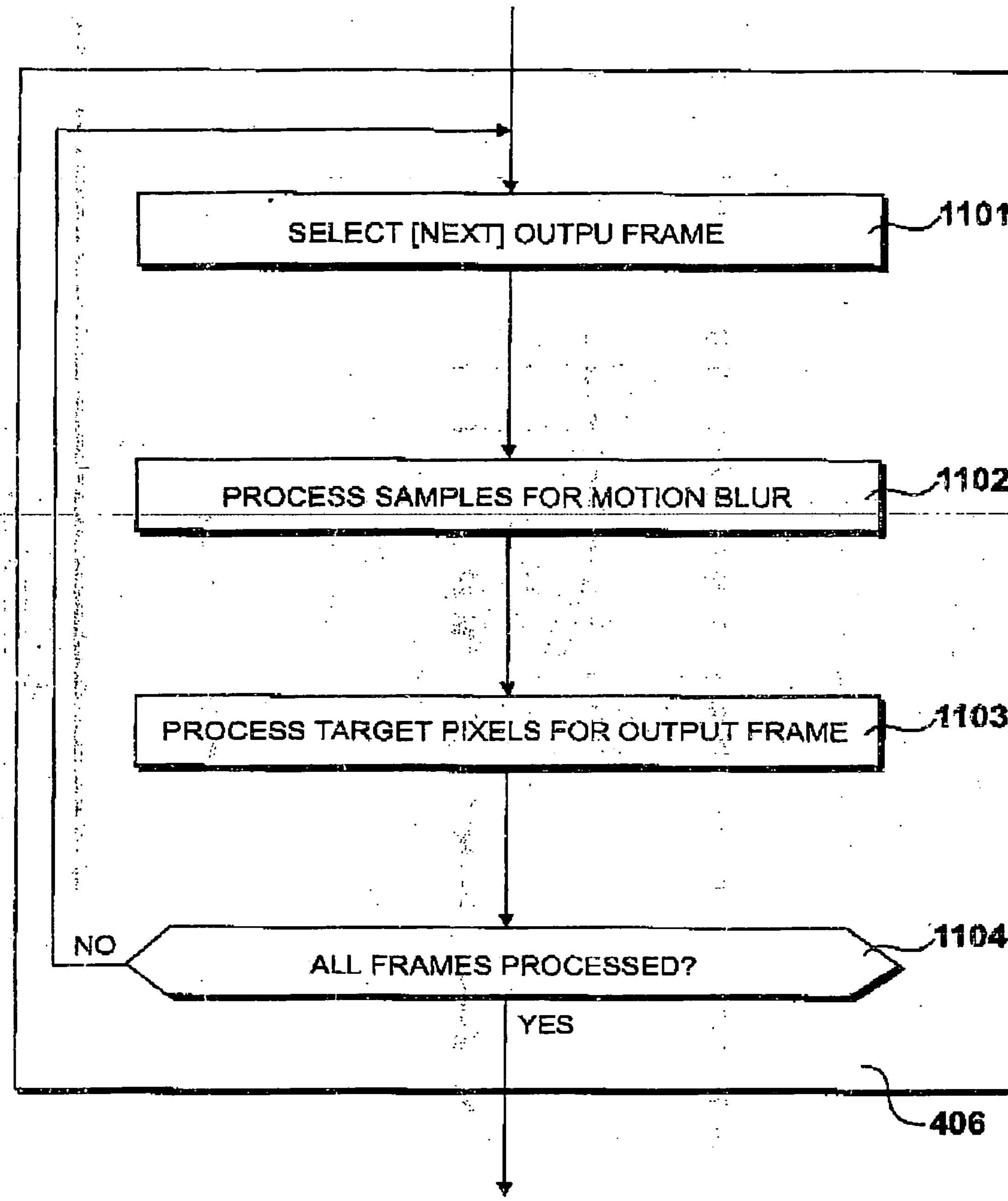

709
704
1002
1001
703
302
301
807
1001
1002
1003
OBJECT 301 → PROPERTIES → PATH → [05:05:10 - 10:05:12]
OBJECT 704 → PROPERTIES → PATH → [TRACK] [x,y]
...
OBJECT n → PROPERTIES → PATH → [DEFINE]

Figure 10



SELECT [NEXT] OUTPU FRAME
1101
PROCESS SAMPLES FOR MOTION BLUR
1102
PROCESS TARGET PIXELS FOR OUTPUT FRAME
1103
NO
ALL FRAMES PROCESSED?
1104
YES
406

Figure 11

2034-P585-GB

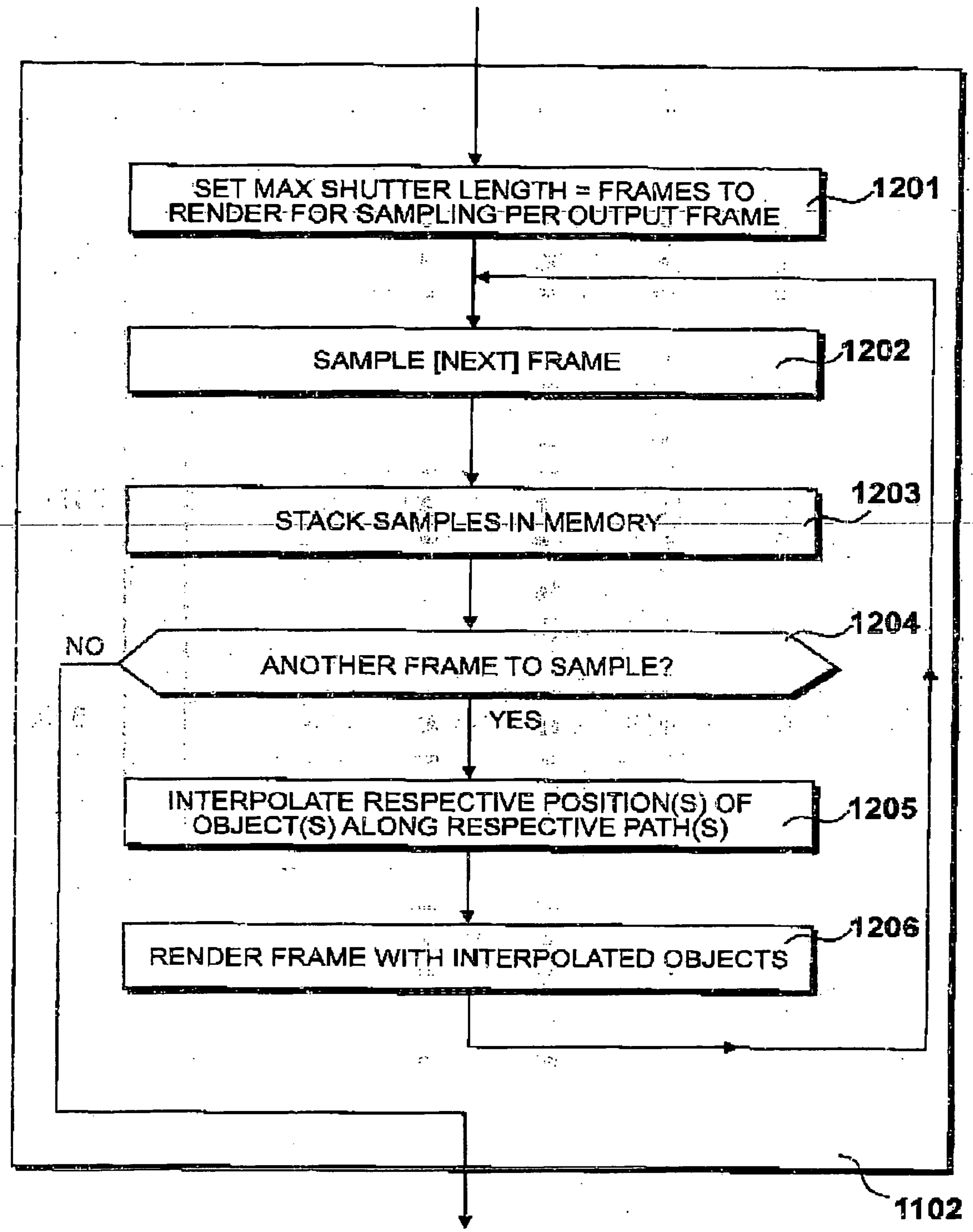


Figure 12

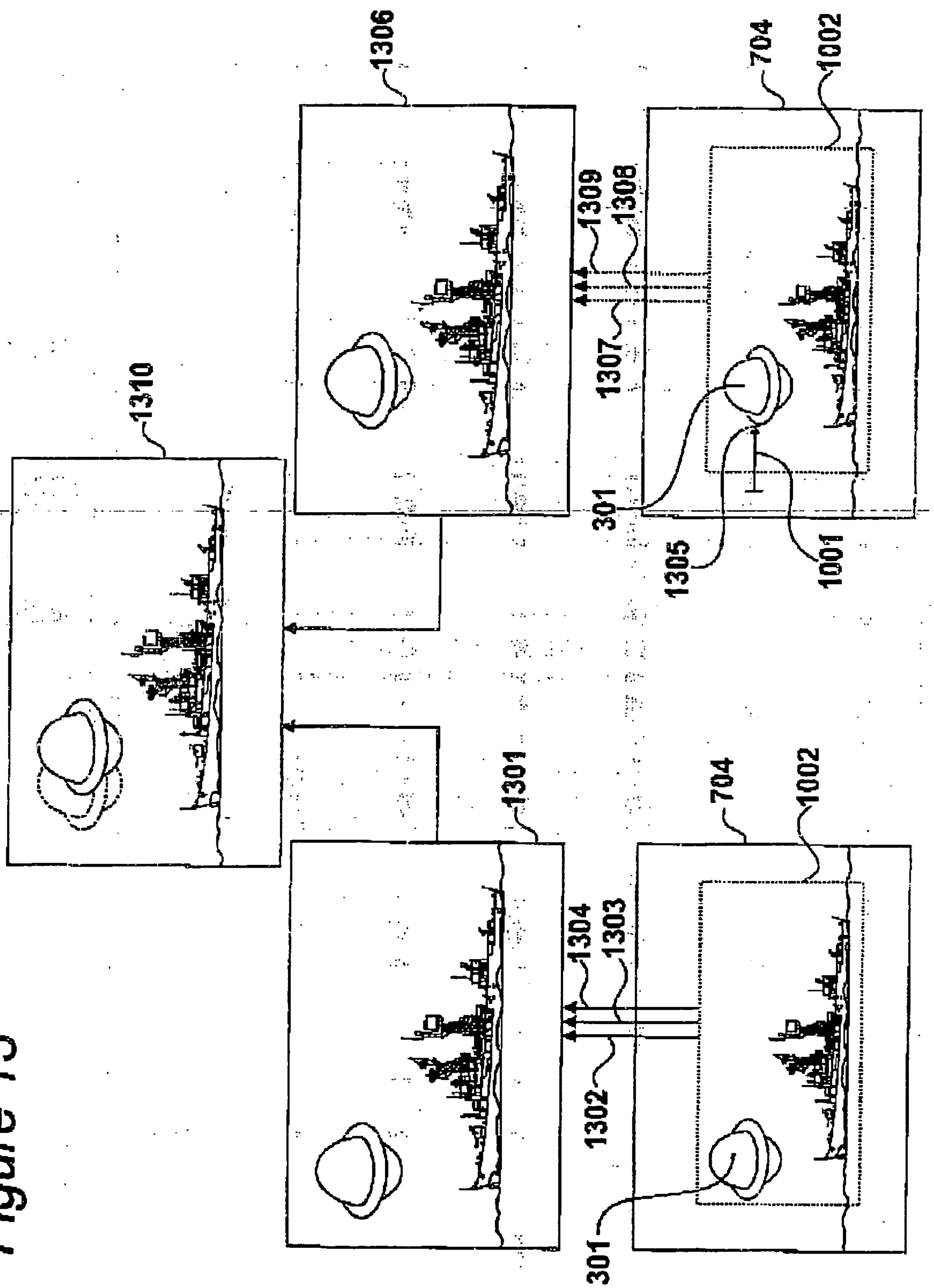
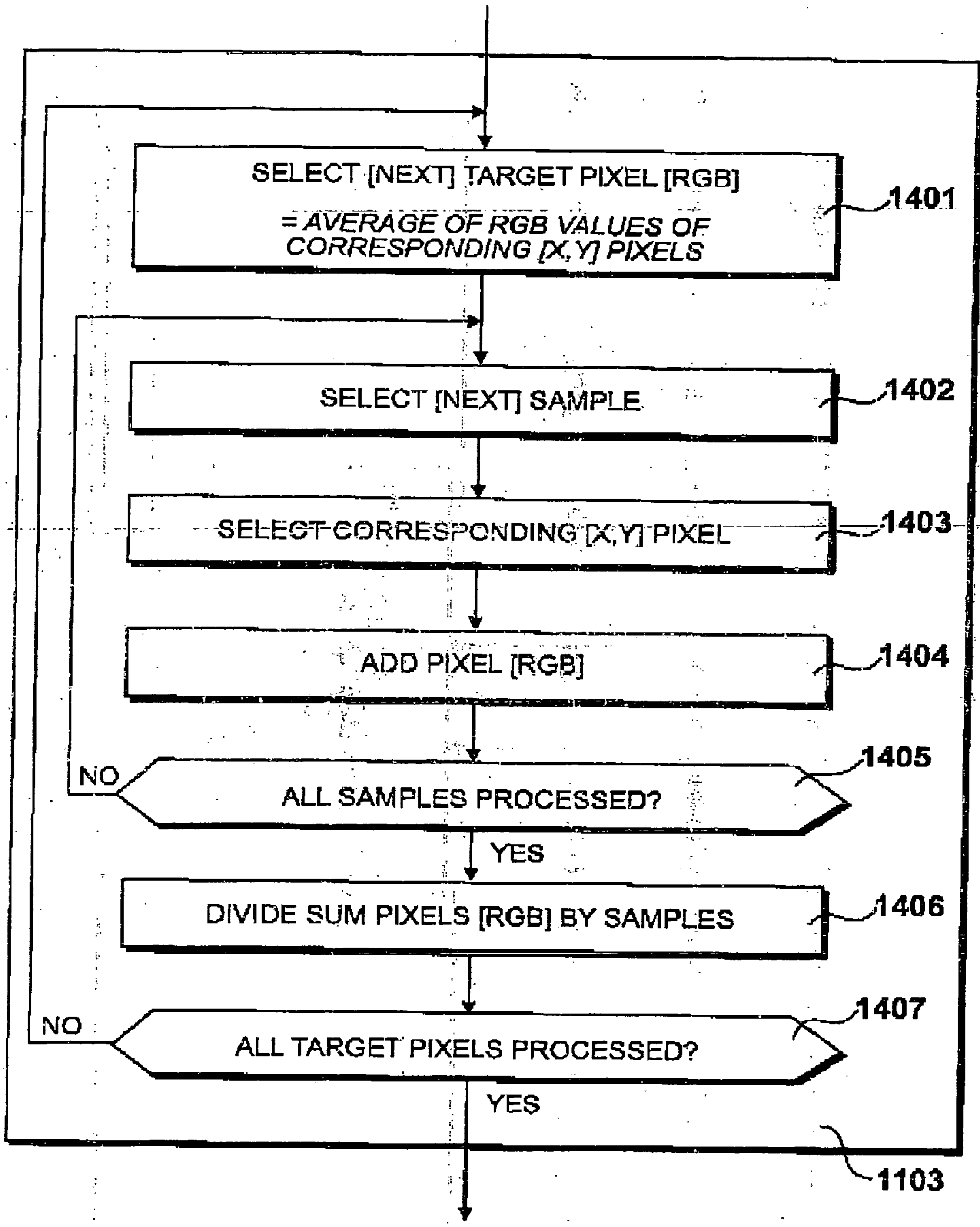

1306
704
1002
1309
1308
1307
1310
301
1305
1001
1301
704
1002
1304
1303
1302
301

Figure 13



SELECT [NEXT] TARGET PIXEL [RGB]
= AVERAGE OF RGB VALUES OF CORRESPONDING [X,Y] PIXELS
1401
SELECT [NEXT] SAMPLE
1402
SELECT CORRESPONDING [X,Y] PIXEL
1403
ADD PIXEL [RGB]
1404
ALL SAMPLES PROCESSED?
1405
NO
YES
DIVIDE SUM PIXELS [RGB] BY SAMPLES
1406
ALL TARGET PIXELS PROCESSED?
1407
NO
YES
1103

Figure 14

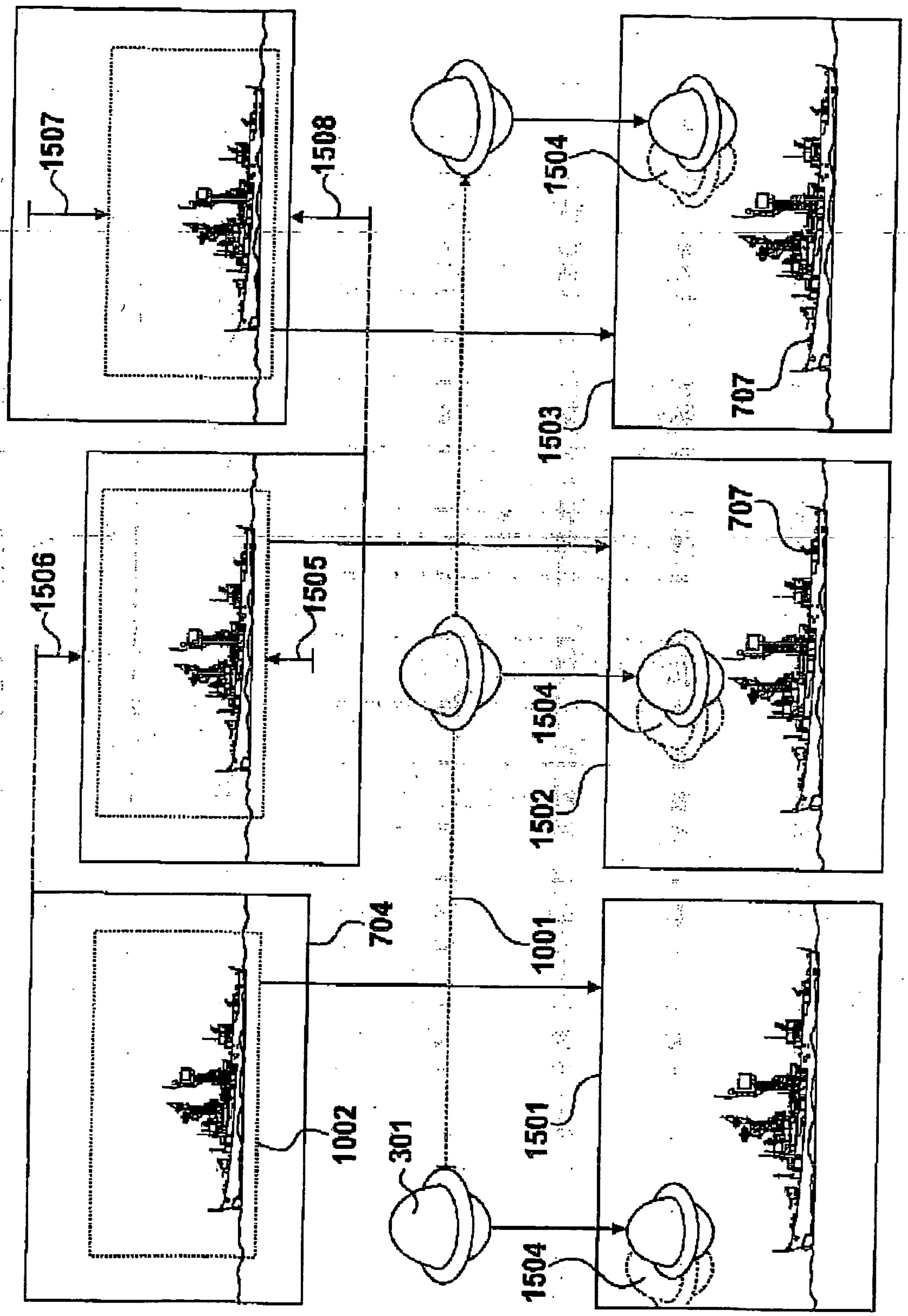

1507
1508
1504
1503
707
1506
1505
707
1504
1502
704
1001
1002
301
1501
1504

Figure 15